A $\mathbb{R}$eal Generalization of Discrete AdaBoost

Richard Nock[1] Frank Nielsen[2]

[1]Université Antilles-Guyane
CEREGMIA, Martinique, France
Richard.Nock@martinique.univ-ag.fr

[2]Sony CS Labs, Inc.
Fundamental Res. Lab., Tokyo, Japan
Frank.Nielsen@acm.org

August 2006

Outline

- Learning (Weak, Strong, and Boosting);
- Discrete and Real AdaBoosts;
- Lifting to $\mathbb{R}$: Margins & our generalization of discrete AdaBoost;
- Properties;
- Experiments;
- Conclusion.

Learning: Setting

Problem

Given: $\boxed{\mathcal{X}}$, *a domain of* ***observations*** *(e.g.* $\mathbb{R}^n, \{0,1\}^n$*);* n *= number of description variables;*

$\boxed{\{-1,+1\}}$, *a set of* ***classes*** *(e.g.* $\{bad, good\}$*,* $\{lose, win\}$*); "*-1*" = negative class; "*$+1$*" = positive class;*

we wish to learn a particular binary relation from $\mathcal{X}$ *to* $\{-1,+1\}$ *(e.g.* $\mathcal{X}$ *= endgame configurations, classes =* $\{lose, win\}$*).*

Framework:

- draw a set of m **examples** $\mathcal{S} = \{(\boldsymbol{x}, y) \in \mathcal{X} \times \{-1,+1\}\}$ according to distribution D (unknown but fixed);
- learn a classifier $H : \mathcal{X} \to \mathbb{R}$, so as to minimize its **true risk with high probability** (+ require learning P-time in relevant parameters).

Learning: Weak/Strong

True risk minimization whp (weak/strong learning)

Let $\epsilon_{D,H} = \mathbf{Pr}_{(\boldsymbol{x},y)\sim D}[sign(H(\boldsymbol{x})) \neq y]$ be the **true risk** of H.

Strong: require $\Pr_{\mathcal{S}\sim D^m}[\epsilon_{D,H} \leq \varepsilon] \geq 1-\delta$ (ε, δ user-fixed);

Weak: require $\Pr_{\mathcal{S}\sim D^m}[\epsilon_{D,H} \leq 1/2-\gamma] \geq \delta'$ (γ, δ' very small: *e.g.* tiny constant, $\approx 1/p(n)$, etc.);

(requirements hold $\forall D$, $\forall 0 < \varepsilon, \delta < 1$).

Strong learning is learning as usual.

Weak learning is the "weakest", as $\epsilon_{D,\text{unbiased coin}} = 1/2, \forall D$.

Fundamental result (Boosting property, Schapire'90)

$\boxed{\text{Weak learning} \implies \text{Strong learning}}$, *i.e.* given algorithm `WL` that weak learns, we can build algorithm `SL` that strong learns with the **sole** access to `WL`.

Empirical Risks & Strong Learning

Sufficient conditions for Strong Learning

Let $\boldsymbol{w}_1$ be the observed distribution on $\mathcal{S}$, and $\epsilon_{\boldsymbol{w}_1,H}$ the **empirical risk** of H: $\epsilon_{\boldsymbol{w}_1,H} = \mathbf{E}_{(\boldsymbol{x},y)\sim\boldsymbol{w}_1}(1_{sign(H(\boldsymbol{x}))\neq y})$ ($1_\pi = 1$ if π is true, and 0 otherwise). Modulo additional conditions,

$$\boxed{\epsilon_{\boldsymbol{w}_1,H} = 0 \ (H \text{ consistent with } \mathcal{S}) \Rightarrow \text{Strong Learning}}$$

The direct minimization of $\epsilon_{\boldsymbol{w}_1,H}$ has drawbacks (not smooth, many potential local minima, Hardness issues).

Solution: minimize a convex, smooth upperbound

Let $\epsilon^{\exp}_{\boldsymbol{w}_1,H} = E_{(\boldsymbol{x},y)\sim\boldsymbol{w}_1}(\exp(-yH(\boldsymbol{x})))$ be the **exponential loss**.

Advantage 1 : $\epsilon^{\exp}_{\boldsymbol{w}_1,H}$ is convex and smooth differentiable.

$\boxed{\text{Advantage 2}}$: $\epsilon_{\boldsymbol{w}_1,H} \leq \epsilon^{\exp}_{\boldsymbol{w}_1,H}$, as $1_{sign(H(\boldsymbol{x}))\neq y} \leq \exp(-yH(\boldsymbol{x}))$.

$\boxed{\textbf{Advantage 3}}$: $\epsilon^{\exp}_{\boldsymbol{w}_1,H}$ takes full advantage that $H(\boldsymbol{x}) \in \mathbb{R}$.

Problem

Fix $H = H_T$ *a linear combination of* T *classifiers* (h_t) *from* WL*:* $H_T(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})$*. How can we fit* H_T *to the min of* $\epsilon^{\mathrm{exp}}_{\boldsymbol{w}_1, H_T}$ *?*

The solution is a stagewise approach:

Brute force

for $t = 1, 2, ..., T$

1. $h_t \leftarrow \mathtt{WL}(\mathcal{S}, \boldsymbol{w}_1)$;
2. $\alpha_t \leftarrow \arg\min_{\alpha \in \mathbb{R}} \epsilon^{\mathrm{exp}}_{\boldsymbol{w}_1, H_{t-1} + \alpha h_t}$;

return $H_T = \sum_{t=1}^{T} \alpha_t h_t$;

(Friedman & al.'00)

AdaBoost

for $t = 1, 2, ..., T$

1. $h_t \leftarrow \mathtt{WL}(\mathcal{S}, \boxed{\boldsymbol{w}_t})$;
2. $\alpha_t \leftarrow \arg\min_{\alpha \in \mathbb{R}} \epsilon^{\mathrm{exp}}_{\boldsymbol{w}_t, \alpha h_t}$;
3. $w_{t+1,i} \leftarrow \frac{w_{t,i} \exp(-y_i \alpha_t h_t(\boldsymbol{x}_i))}{Z_t}$; $(\forall (\boldsymbol{x}_i, y_i) \in \mathcal{S})$

return $H_T = \sum_{t=1}^{T} \alpha_t h_t$;

Discrete and Real AdaBoosts (FS'97, KW'99, SS'99)

Let $h_t : \mathcal{X} \rightarrow \mathbb{B} \subseteq \mathbb{R}$. AdaBoost comes with two different flavors, depending on $\mathbb{B}$... and all **are** boosting algorithms.

Real AdaBoost

for $t = 1, 2, ..., T$

1 $h_t \leftarrow \mathtt{WL}(\mathcal{S}, \boldsymbol{w}_t)$;

2 $\boxed{\alpha_t \leftarrow \arg\min_{\alpha \in \mathbb{R}} \epsilon^{\mathrm{exp}}_{\boldsymbol{w}_t, \alpha h_t}}$;

3 $w_{t+1,i} \leftarrow \frac{w_{t,i} \exp(-y_i \alpha_t h_t(\boldsymbol{x}_i))}{Z_t}$; $(\forall (\boldsymbol{x}_i, y_i) \in \mathcal{S})$

return $H_T = \sum_{t=1}^{T} \alpha_t h_t$;

Discrete AdaBoost

for $t = 1, 2, ..., T$

1 $h_t \leftarrow \mathtt{WL}(\mathcal{S}, \boldsymbol{w}_t)$;

2 $\boxed{\alpha_t \leftarrow {}^1\!/_2 \log\left((1 - \epsilon_{\boldsymbol{w}_t, h_t})/\epsilon_{\boldsymbol{w}_t, h_t}\right)}$;

3 $w_{t+1,i} \leftarrow \frac{w_{t,i} \exp(-y_i \alpha_t h_t(\boldsymbol{x}_i))}{Z_t}$; $(\forall (\boldsymbol{x}_i, y_i) \in \mathcal{S})$

return $H_T = \sum_{t=1}^{T} \alpha_t h_t$;

+ *Any* $\mathbb{B} \subseteq \mathbb{R}$;
- complexity (no closed form for [2]), numerical stability (weights), outside the boosting regime (risk)

+ Straightforward to implement, “best off the shelf classifier in the world”;
- restricted to $\mathbb{B} = \{-1, +1\}$;

Lifting to $\mathbb{R}$: Margins (I)

In a $\mathbb{R}$eal-world, prediction $H_T(\boldsymbol{x}) \in \mathbb{R}$ may be interpreted as:

- a class ($sign(H_T(\boldsymbol{x}))$);
- a confidence in the classification ($|H_T(\boldsymbol{x})|$).

Ideally, the **optimal** classifier gives (i) the right class with (ii) the largest confidence (*i.e.* $+\infty$ when Bayes optimum is zero).
Ex: **logit** prediction, $H(\boldsymbol{x}) = \log \frac{\Pr[y=+1|\boldsymbol{x}]}{\Pr[y=-1|\boldsymbol{x}]}$ (Friedman & al.'00).

What we want is a criterion $\ell_H((\boldsymbol{x}, y))$ integrating both the sign and the confidence, instead of just $\ell_H((\boldsymbol{x}, y)) = 1_{sign(H(\boldsymbol{x}))\neq y}$.

Margin $\ell_H((\boldsymbol{x}, y))$

A **margin** (of H on $(\boldsymbol{x}, y)$) satisfies four requirements:

1. it is a function of $yH(\boldsymbol{x})$;
2. it is monotonic increasing;
3. it is $\in [-1, +1]$;
4. negative iff $1_{sign(H(\boldsymbol{x}))\neq y} = 1$.

Margin of H_T on example $(\boldsymbol{x}, y)$

$$\ell_{H_T}((\boldsymbol{x}, y)) = \frac{\exp(yH_T(\boldsymbol{x})) - 1}{\exp(yH_T(\boldsymbol{x})) + 1} \quad \begin{cases} \leq 1 & \text{(good label, } \infty \text{ confidence)} \\ \geq -1 & \text{(bad label, } \infty \text{ confidence)} \end{cases}$$

Logit brings $\ell_{H_T}((\boldsymbol{x}, y)) = y(2\Pr[y = +1|\boldsymbol{x}] - 1)$ (Friedman & al.'00).

Definition

Let $-1 \leq \theta \leq 1$. The **margin error**, $\nu_{\boldsymbol{w}_1,H_T,\theta}$, is the proportion of examples whose margin does not exceed θ:

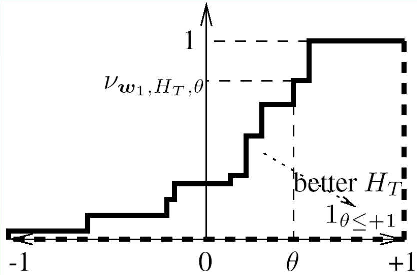
$$\nu_{\boldsymbol{w}_1,H_T,\theta} = \mathbf{E}_{(\boldsymbol{x},y)\sim\boldsymbol{w}_1}(1_{\ell_{H_T}((\boldsymbol{x},y))\leq\theta})$$

We have $\boxed{\epsilon_{\boldsymbol{w}_1,H} \leq \nu_{\boldsymbol{w}_1,H_T,0}}$.

H_T is as good as:

- $\nu_{\boldsymbol{w}_1,H_T,\theta}$ is small,
- for any θ.

  (*i.e.* the distribution of margins $\rightarrow$ curve $1_{\theta\leq+1}$)



Lifting to $\mathbb{R}$: Our Real Generalization of AdaBoost

Principle

"Forget" the direct minimization of the **exponential loss**. Rather focus on the **margin error**.

Real AdaBoost

for $t = 1, 2, ..., T$

1 $h_t \leftarrow \mathtt{WL}(\mathcal{S}, \boldsymbol{w}_t)$;

2 $\alpha_t \leftarrow \arg\min_{\alpha \in \mathbb{R}} \epsilon^{\exp}_{\boldsymbol{w}_t, \alpha h_t}$;

3 $w_{t+1,i} \leftarrow w_{t,i} \exp(-y_i \alpha_t h_t(\boldsymbol{x}_i))/Z_t, \forall i$;

return $H_T = \sum_{t=1}^T \alpha_t h_t$;

AdaBoost$_\mathbb{R}$ (Our generalization)

for $t = 1, 2, ..., T$

1 $h_t \leftarrow \mathtt{WL}(\mathcal{S}, \boldsymbol{w}_t)$;

2 $\alpha_t \leftarrow \frac{1}{2h_t^\star} \log \frac{1+\mu_t}{1-\mu_t}$;

3 $w_{t+1,i} \leftarrow w_{t,i} \times \frac{1-(\mu_t y_i h_t(\boldsymbol{x}_i))/h_t^\star}{1-\mu_t^2}, \forall i$;

return $H_T = \sum_{t=1}^T \alpha_t h_t$;

$h_t^\star = \max_{(\boldsymbol{x},y)\in\mathcal{S}} |h_t(\boldsymbol{x})| \in \mathbb{R}^+$

$\mu_t = \mathbf{E}_{(\boldsymbol{x},y)\sim \boldsymbol{w}_1}(y h_t(\boldsymbol{x})/h_t^\star) \in [-1, +1]$

(note: $y h_t(\boldsymbol{x})/h_t^\star = \ell_{h_t}((\boldsymbol{x}, y))$ **is also a margin** on example $(\boldsymbol{x}, y)$).

Properties (I): AdaBoost$_{\mathbb{R}}$ boosts **all** margins

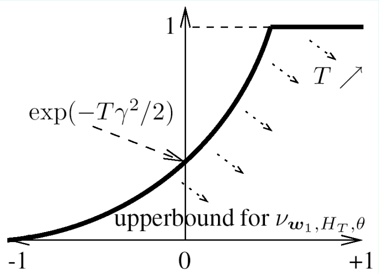
Recall that $\nu_{\boldsymbol{w}_1, H_T, \theta}$ counts the % of examples with margin $\leq \theta$.

Theorem

$\forall \mathbb{B} \subseteq \mathbb{R}$, $\boxed{\forall \theta \in [-1, +1]}$, *the margin error of* H_T *satisfies:*

$$\nu_{\boldsymbol{w}_1, H_T, \theta} \leq \left(\frac{1+\theta}{1-\theta}\right) \times \exp\left(-\frac{1}{2}\sum_{t=1}^{T} \mu_t^2\right)$$

- Suppose $\boxed{\forall t \geq 1, |\mu_t| \geq \gamma}$ for some small $\gamma > 0$.
- Then $\nu_{\boldsymbol{w}_1, H_T, \theta} \leq f(\theta) \times \exp(-\frac{T\gamma^2}{2})$.
- As $T \nearrow$, the lhs $\rightarrow 1_{\theta \leq +1}$.



Recall that $(\forall t \geq 1, |\mu_t| \geq \gamma) \Rightarrow \left(\nu_{\boldsymbol{w}_1, H_T, \theta} \leq f(\theta) \times \exp(-\frac{T\gamma^2}{2})\right)$.

- Run AdaBoost$_{\mathbb{R}}$ with $T = \Omega\left(\frac{1}{\gamma^2} \log \frac{f(\theta)}{\min_i w_{1,i}}\right)$;
- we obtain $\boxed{\nu_{\boldsymbol{w}_1, H_T, \theta} = 0}$;
- use with $\theta = 0$ to prove $\epsilon_{\boldsymbol{w}_1, H_T} = 0$;
- (+more material) $\Rightarrow$ AdaBoost$_{\mathbb{R}}$ **is a boosting algorithm**.

Weak Learning is in the assumption $|\mu_t| \geq \gamma$

	$\mathbb{B} = \{-1, +1\}$	$\mathbb{B} \subseteq \mathbb{R}$
Random	Unbiased coin	Uniform $\in [-b, +b]$
Satisfies	$\epsilon_{\boldsymbol{w}_t, .} = {}^1/_2$	$\mu_t = 0$
Weak Learning	$\left\{\begin{array}{lll} \epsilon_{\boldsymbol{w}_t, h_t} & \leq & {}^1/_2 - {}^\gamma/_2 \\ \epsilon_{\boldsymbol{w}_t, h_t} & \geq & {}^1/_2 + {}^\gamma/_2 \end{array}\right.$	$\|\mu_t\| \geq \gamma$
Property	**WL for $\mathbb{B} \subseteq \mathbb{R}$ generalizes WL for $\mathbb{B} = \{-1, +1\}$**	

For simplicity, we do not plug "$\Pr_{\mathcal{S} \sim D^m}[.] \geq \delta'$": it would not change anything.

AdaBoost$_{\mathbb{R}}$ **is** a generalization of Discrete AdaBoost (perfect match if $\mathbb{B} = \{-1, +1\}$).

Compared to other Real AdaBoosts:

- **all** the algorithm is in closed form (no approximation = no complexity penalty);
- it works properly even on **limit regimes**:
  - when h_t takes ∞ values (*e.g.* DT + logit at the leaves); yields *e.g.* $\mu_t = \sum_{i:|h_t(\boldsymbol{x}_i)|=\infty} w_{1,i} sign(y_i h_t(\boldsymbol{x}_i))$;
  - when $\epsilon_{\boldsymbol{w}_t, h_t} \rightarrow 0, 1$ (no weight change !);
- the computation of the leveraging coefficients (α_t) can be **delayed** towards the end of learning (reduces numerical instabilities);

Weight modification rule

$\mathbb{B} = \{-1, +1\}$ Perhaps the most popular fact about (Discrete Ada)Boosting is that examples **correctly** (resp. badly) classified by h_t get their weight **decreased** (resp. increased) (holds when $\epsilon_{\boldsymbol{w}_t, h_t} \leq 1/2$; otherwise, reverses the polarity).

$\mathbb{B} = \mathbb{R}$ In AdaBoost$_{\mathbb{R}}$, examples that have their weight **decreased** are those for which:

$$\ell_{h_t}((\boldsymbol{x}, y)) \geq \mu_t$$

The weak classifier's "local margin" exceeds its average margin (holds when $\mu_t \geq 0$; otherwise, reverses the polarity).

Experiments (I): Experimental setting

We pick 25 domains, most from the UCI repository.

- 10-fold stratified cross-validation, $T \in \{10, 50\}$;
- WL returns monomials with fixed length (Rank-1 DT with fixed depth, Nock'02);
- we compare three algorithms:
  1. Discrete AdaBoost (Freund & Schapire'97),
  2. AdaBoost$_{\mathbb{R}}$,
  3. Real AdaBoost (Kivinen & Warmuth'99, Schapire & Singer'99),
     with α_t approximated up to relative error $\leq 10^{-6}$, and using results from (Nock & Nielsen'06) to make the search faster;

Execution time

The implementations of Real AdaBoost and AdaBoost$_{\mathbb{R}}$ use the same routines (same optimization).

The execution time for AdaBoost$_{\mathbb{R}}$ was smaller by **orders of magnitude**.

Experiments (II): General results

(see paper for details)

	$T = 10$			$T = 50$		
	D	U	T	D	U	T
#**best**	7	**11**	9	7	**15**	6
#second	9	6	5	9	4	5
#*worst*	9	8	11	9	6	14

D = Discrete AdaBoost
U = AdaBoost$_{\mathbb{R}}$
T = Real AdaBoost

As T increases, AdaBoost$_{\mathbb{R}}$ tends to become the winner.

Hard domains

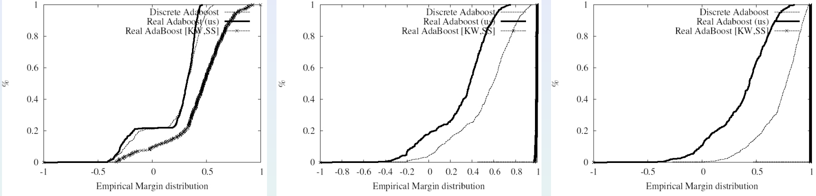
On harder simulated domains (class/attribute noise, irrelevant attributes), AdaBoost$_{\mathbb{R}}$ becomes the clear winner as T increases.

We think that this might be due to our **gentler** weight update rule.

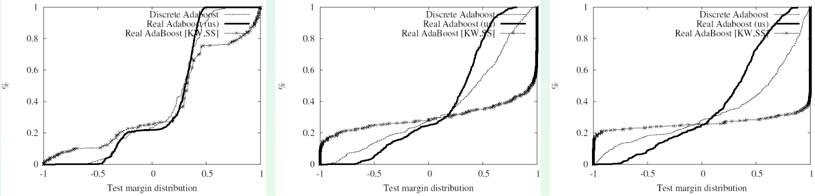
Experiments (III): Margins

$T = 10$	$T = 100$	$T = 200$

cumulative distributions of margins on **training**



cumulative distributions of margins on **testing**



$r = 6$ litterals per rule. Recall that $\nu_{\mathbf{w}_1, H_T, \theta}$ should be as small as possible, all the more for $\theta < 0$.

Conclusion

- Since we deal with real-valued predictions, learning should take into account both the sign **and** the confidence in the model:

Strong: require $\Pr_{\mathcal{S}\sim D^m}[\nu_{D,H,\theta} \leq \varepsilon] \geq 1-\delta$ ($\varepsilon, \delta, \theta$ user-fixed);

Weak: require $\Pr_{\mathcal{S}\sim D^m}[\mu_t \geq \gamma] \geq \delta'$ (γ, δ' very small: *e.g.* tiny constant, $\approx 1/p(n)$, etc.);

$\Rightarrow$ What happens ?

- Integrate Bayes rule in the bounds, and investigate convergence / stability.
- Multiclass extensions.
- etc.

Thank you for your attention

- Acknowledgements: **Sony CSL Tokyo** for a visiting grant during which part of this work was done (R. Nock).
- References:

1. Schapire'90: "The Strength of Weak Learnability", **Machine Learning Journal**;
2. Friedman & al.'00: "Additive Logistic Regression: a Statistical View of Boosting", **Annals of Statistics**;
3. Nock'02: "Inducing Interpretable Voting Classifiers without trading Accuracy for Simplicity: Theoretical results, Approximation algorithms, and Experiments", **Journal of Artificial Intelligence Research**;
4. Freund & Schapire'97: "A Decision-Theoretic Generalization of On-line learning and an application to Boosting", **Journal of Computer and System Sciences**;
5. Kivinen & Warmuth'99: "Boosting as Entropy Projection", **ACM Int. Conference on Computational Learning Theory**;
6. Schapire & Singer'99: "Improved Boosting algorithms using Confidence-rated Prediction", **Machine Learning Journal**;
7. Nock & Nielsen'06: "On Weighting Clustering", **IEEE Transactions on Pattern Analysis and Machine Intelligence**;